MEMORANDUM
RM-5564-NRL
FEBRUARY 1968

AD 666830

A METHOD OF OBJECTIVE CONTOUR CONSTRUCTION

F. W. Murray

PREPARED FOR:
OFFICE OF NAVAL RESEARCH

The RAND Corporation
SANTA MONICA • CALIFORNIA

MEMORANDUM
RM-5564-NRL
FEBRUARY 1968

A METHOD OF OBJECTIVE CONTOUR CONSTRUCTION

F. W. Murray

This research is supported by the Office of Naval Research under Contract No. N00014-67-C-0101, Reqn. 00178-6-006280. RAND Memoranda are subject to critical review procedures at the research department and corporate levels. Views and conclusions expressed herein are nevertheless the primary responsibility of the author, and should not be interpreted as representing the official opinion or policy of the Office of Naval Research or of The RAND Corporation.

The RAND Corporation
1700 MAIN ST • SANTA MONICA • CALIFORNIA • 90406

PREFACE

Many numerical models used in meteorology and other branches of geophysics produce large numbers of fields of data that can best be studied by constructing graphical contours of the fields. This is particularly true of the cumulus dynamics program developed at RAND under sponsorship of the Naval Research Laboratory. A given computer run of this program may produce fields of five or more variables at upwards of 100 time steps. If any substantial fraction of the fields is to be contoured, it must be done mechanically.

Contour programs have been written for use in most numerical weather prediction facilities as well as in some research agencies. Unfortunately, they are all specialized as regards either the fields contoured or the equipment on which the contouring was done, or both. When attempts to find a ready-made contour program suitable for the RAND cumulus dynamics model failed, it was decided to develop one along the simplest possible lines that would give reasonably accurate and esthetically pleasing results. The method so developed is described herein.

ABSTRACT

Problems of graphical display of a dependent variable as a function of two independent variables are discussed. A procedure for objective construction of contours is described, and the program logic is presented. Examples of contours of fields developed by a cumulus dynamics program and produced by a General Dynamics S-C4060 are shown.

CONTENTS

I. INTRODUCTION

In many geophysical problems the values of a dependent variable are best portrayed as a function of two independent variables. This is most conveniently done by drawing contours of the dependent variable on a grid representing the independent variables. Ordinarily the value of the dependent variable is known for only a limited number of pairs, and so some form of interpolation is necessary. The most direct procedure is to plot the known values on a grid, and then to draw the contours by hand, interpolating by eye. Inevitably some smoothing results, the amount being determined subjectively and often varying from one part of the chart to another. After skill has been acquired by considerable practice, this procedure can result in an esthetically pleasing chart that will yield the value of the dependent variable at any point in the field to a degree of accuracy consistent with the accuracy and density of the original data.

This procedure, however, is time-consuming, and moreover requires the acquisition of a specialized skill. When the data concerned are the output of a computer program, there are frequently many charts to plot and analyze, and the investigator is often deterred by the sheer magnitude of the task. Some help may be afforded by programming the computer to print the values of the dependent variable in their appropriate spatial relationships, thereby eliminating the tedious step of extracting values from a table and plotting them on a grid. This procedure, however, does nothing to reduce the more difficult and specialized task of analysis. In some instances the computer has in addition been allowed to interpolate among the grid values and the printer has been programmed to strike characters in the appropriate location, so that the analyst need only connect like characters to draw the contours. The results, however, are at best crude and inelegant.

A better method is to use a specialized graphical-output device. Two categories of these devices are in general use: mechanical and electronic. With the former the computer determines by interpolation the coordinates of successive points on a contour at close intervals, and the output device thereupon causes a pen to move accordingly across

a sheet of paper. With the latter the computer also determines the coordinates of successive points, but the contour segment is then displayed electronically on a screen that may be photographed for permanent record. Both usually have means of producing labels and other alphanumeric information. The essential difference from the point of view of the programmer is that with the mechanical curve-follower it is necessary that all the points of a given contour be sequentially determined before the next contour is considered, whereas the electronic device permits the determination of all contour segments within a given area (say one grid square) before the next area is considered. The latter system leads to significantly fewer logical complexities. On the other hand, the inertia of the pen arm of a mechanical device may cause the contour to be smoothly curved (which is in accordance with our notion of how many variables behave in nature), whereas the electronic device usually displays a series of straight-line segments with angular joinings, unless one resorts to complicated programming. In the present study the General Dynamics S-C4060, an electronic device, was used.

II. INTERPOLATION

If the dependent variable is conceived to be a smooth function of the independent variables, some form of surface fitting is indicated for interpolation among grid points. A quadratic surface can be fitted exactly to six grid points, but since it is impossible to select six points symmetrically distributed about a grid square, it is better to use twelve or sixteen points (see Fig. 1) and to fit the surface by the method of least squares. Of course with these many grid points a higher-order grid surface could be fitted, but the quadratic surface is likely to give the smoothest curves, and without further knowledge of the nature of the dependent variable, no particular surface can be definitely said to be better than another. One drawback of arrays such as those of Fig. 1 is that they do not lend themselves to interpolation within grid squares adjacent to the boundaries. Another disadvantage of the surface-fitting method is that although the resulting curves are smooth within the central grid square, there is no assurance that they will join smoothly (or even join at all) with curves in an adjacent grid square, which are computed from a different set of grid points.

If we are willing to accept angular joinings at grid lines and straight contour segments between, we can effect a great simplification by using linear interpolation along the edges rather than surface fitting.* If the grid mesh is sufficiently fine and the dependent variable is reasonably well-behaved, it is possible to generate quite acceptable contours by this method, even though the finished chart is less pleasing in appearance than one drawn by a skillful analyst.

The method chosen, therefore, is to scan all the grid squares once for each value corresponding to a contour to be drawn, to find by linear interpolation all the points where the contour intersects the edge of the grid square, and to instruct the graphical-output device to connect these points with straight-line segments. The procedure for doing this and the resolution of certain ambiguities that arise are described below.

---

*Double linear interpolation carried out in the interior of a grid square is equivalent to fitting a quadratic surface without the cross-product term. The resulting contour segments are portions of hyperbolas.

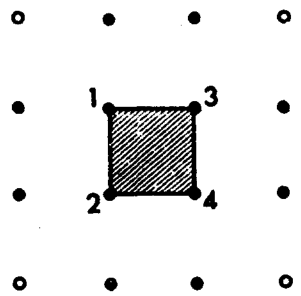

Fig. 1 -- Grid-point arrays suitable for interpolation within shaded square. Solid dots are for 12-point array; circles are added for 16-point array.

III. ENUMERATION OF POSSIBLE CASES

Let the following symbols have the meanings shown:

G = value of the dependent variable.

C = value of the contour under consideration.

i or j = subscript designating a grid point. By analogy with a map, the subscript has the value 1, 2, 3, or 4, respectively, for the "northwest," "southwest," "northeast," and "southeast" corners of a grid square. (See numbers associated with the shaded square in Fig. 1.)

$\delta_i = G_i - C.$

In general, if $\delta_i = 0$, the contour will pass through the grid point designated by i. (Exceptions to this rule will be discussed later.) If i and j represent two grid points on one edge of a grid square, and δ_i and δ_j have opposite signs, the contour will intersect that edge. Linear interpolation places the point of intersection at a distance $(C - G_i)/(G_j - G_i)$, in units of mesh length, from the point i. Continuity requires that if one edge is so intersected, the contour must also intersect one of the other three edges or else go through one of the two grid points not on the first edge. The contour is assumed to follow a straight line between the two endpoints so determined. The four values δ_1, δ_2, δ_3, and δ_4 give all the information that is available or (if certain conventions are adopted) necessary to define the course of the contour within the grid square.

Much useful information is given merely by the signs of δ_1, δ_2, δ_3, and δ_4. The type of contour configuration within a grid square, though not necessarily its exact description, is given uniquely by these four bits (actually ternary digits) of information. The number of order-preserving combinations of m members in which each member is one of n symbols is one greater than the largest m-digit number to number base n; i.e., it is n^m. Since in the present instance $m = 4$, representing the four corners of the grid square, and $n = 3$, representing the symbols -,

0, and +, the number of combinations is 81. Since $(C - G_i)/(G_j - G_i) = \delta_i/(\delta_i - \delta_j)$, it is obvious that if all the signs of the quantities δ_1, δ_2, δ_3, and δ_4 are reversed, the course of the contour within the grid square remains unchanged. Thus about half of the 81 cases are redundant and need not be considered separately. The remaining 41 cases are illustrated in Fig. 2.

In this figure the sign of δ_i is shown beside the grid point. The square in row A and column ε (designated Aε), having each δ_i equal to zero, is unaffected by a change of sign. If Aδ were illustrated, it would be the negative of Aζ, Aγ would be the negative of Aη, and so on. The 40 omitted cases are symmetrical with the 40 nonzero cases illustrated.

The contours are shown schematically in Fig. 2 as heavy lines. When a contour intersects an edge, it is shown in this illustration as intersecting the midpoint, although in practice its intersection would be determined by the magnitude of $\delta_i/(\delta_i - \delta_j)$. In four cases (Aη, Bδ, Cε, and Dβ) the ratio $\delta_i/(\delta_i - \delta_j)$ has the form 0/0, but it is arbitrarily assumed to take the value 1/2. Such ambiguities are discussed more fully in the next chapter.

In order to construct contours as illustrated in Fig. 2, the program must compute raster numbers for the two coordinates of each endpoint of each contour segment. For the x-coordinates there are five conditions, including the arbitrary case just mentioned, each requiring its own method of computation. They are:

1. The endpoint lies on the left-hand edge.
2. The endpoint is determined by $\delta_1/(\delta_1 - \delta_3)$.
3. The endpoint is determined by $\delta_2/(\delta_2 - \delta_4)$.
4. The endpoint lies on the right-hand edge.
5. The endpoint lies midway on the upper or lower edge.

The corresponding conditions for the y-coordinate are:

1. The endpoint lies on the upper edge.
2. The endpoint is determined by $\delta_1/(\delta_1 - \delta_2)$.
3. The endpoint is determined by $\delta_3/(\delta_3 - \delta_4)$.
4. The endpoint lies on the lower edge.
5. The endpoint lies midway on the right-hand or left-hand edge.

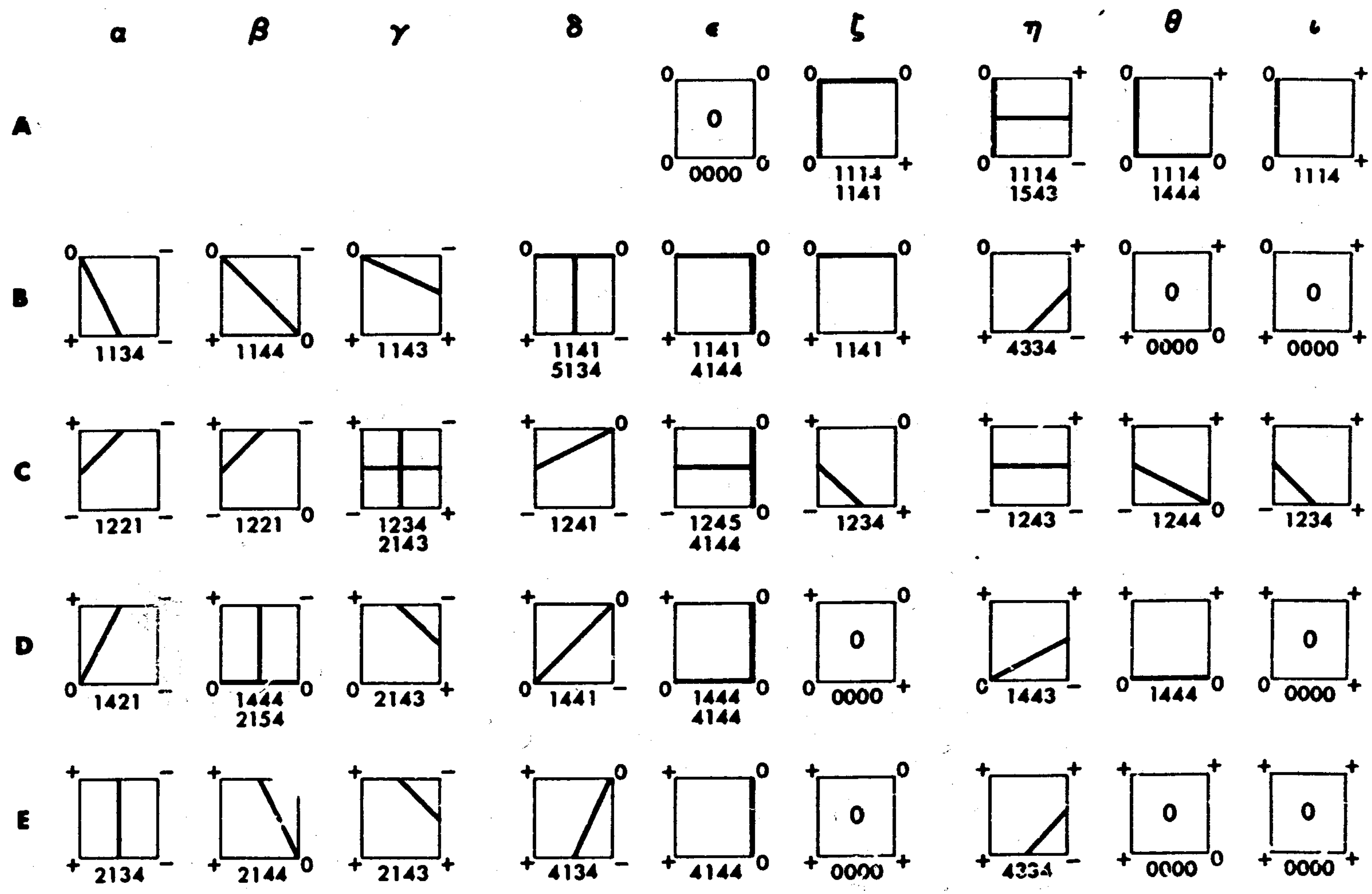

Fig. 2 -- Possible contour configurations in a grid square.

Although for both coordinates conditions 1 and 4 are special cases of 2 and 3, it is desirable to consider them separately. Thus the general nature of the contour configuration can be concisely expressed by a four-digit number, $I_1\ J_1\ I_2\ J_2$, where I_1 and J_1, each running from 1 to 5, refer respectively to the x- and y-coordinates of one endpoint and I_2 and J_2 similarly represent the other endpoint. This representation corresponds exactly to the schematic graphical representation, but is more convenient to use in blocking out the program. The digital representation is shown in Fig. 2 below the grid squares. Some cases have two contour segments and some have none. If there are none, the graphical representation is a zero in the grid square, and the digital representation is 0000.

IV. RESOLUTION OF AMBIGUITIES

In many instances (e.g., $B\beta$, $C\alpha$, $C\eta$, $D\alpha$) the placement of the contour is unequivocal. In other cases, there are several logical possibilities, and the choice made was more or less arbitrary. Occasionally information from adjoining grid squares is needed to make the best choice, but it was felt that the improvement in pattern gained by using such information is not worth the extra effort entailed.

The most ambiguous case is that of the saddle point, $C\gamma$. On the basis of the available data, i.e., the value of G at the four corner points, any of the three configurations shown in Fig. 3 is possible. If the contours intersect the edges near their midpoints, there is no logical basis for choice between configurations I and II, leaving III as the best choice, except for the unaccountable aversion many analysts have to drawing crossed contours. (It can be shown that for every saddle point there exists a contour value that must have an intersection.) If, however, the contours intersect the edges near one or more grid points, configuration III is not necessarily the most logical choice, and there may be a rational basis for a choice between configurations I and II. The various possibilities can all be tested by appropriate programming, and suitable choices can be made objectively, leaving but a small residual ambiguity; however, in view of the complexities introduced by this procedure and the relative infrequency of occurrence of saddle points, it was felt that such programming was not justified. Accordingly, a completely arbitrary decision was made: configuration III was chosen for all saddle points.

Other ambiguous cases all involve a value of zero for δ_i. In cases such as $B\eta$ the value of zero was ignored on the assumption that if it were of any consequence it would be better taken into account in an adjacent grid square. Cases such as $B\delta$ occur mainly on the border. In this configuration the endpoint of the interior contour segment could logically lie anywhere on the zero edge, but without knowledge of the course of the contour segment connecting with the other endpoint and lying in the adjoining grid square, the most reasonable choice of a terminator is the midpoint of the zero edge.

Some fields have extensive flat regions (G = constant), and frequently C is selected so that C = G in such a region. A common example is a field in which G is nonzero in only a limited region, in which case C = 0 is a desirable choice for a base contour. Case Aε is repeatedly found in the midst of a flat region. Obviously in this circumstance a contour should not be drawn along the edges of each grid square. Cases such as Aζ, Bζ, and Bδ are found near the boundaries of a flat region. Thus a flat region having the value C will be surrounded by a contour of the same value.

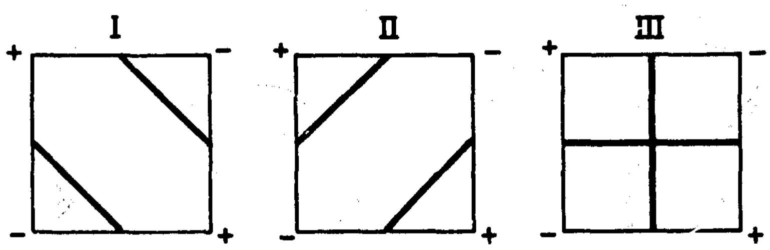

Fig. 3 -- Possible contour configurations for saddle point.

V. PROCEDURE

In practice the entire grid is scanned once for each contour-value desired. For each grid square the signs of δ_1, δ_2, δ_3, and δ_4 are determined. If $\delta_1 > 0$, the appropriate case is selected from rows C, D, and E of Fig. 2. If $\delta_1 < 0$, the sign of each δ_i is changed, and again a selection is made from rows C, D, and E. If, however, $\delta_1 = 0$, the sign of δ_2 is examined. If $\delta_2 > 0$, a choice is made from row B; if $\delta_2 < 0$, all signs are changed, and a choice is made from row B; but if both $\delta_1 = 0$ and $\delta_2 = 0$, δ_3 is examined. For $\delta_3 \neq 0$, one of the cases Aη, Aθ, or Aζ is chosen, and signs are changed if necessary. Finally, if $\delta_1 = \delta_2 = \delta_3 = 0$, case A$\epsilon$ is chosen for $\delta_4 = 0$ and Aζ for $\delta_4 \neq 0$. Once the applicable case has been chosen, it is a simple matter to determine in terms of raster number the coordinates of the two endpoints of each contour segment in the grid square and to require the display device to construct a straight line between pairs of coordinates.

A flow diagram of the procedure is given in Fig. 4. The initial elimination of cases carried out by adjusting the signs of the δ_i is shown in Fig. 4a, and the specification of configuration is shown in the remaining parts of Fig. 4. In this figure a four-character symbol in a rectangle indicates a procedure for finding raster numbers for endpoints of a contour segment. A digit from 1 to 4 in a particular position indicates that the raster number of the coordinate corresponding to that position is to be found in accordance with the digital code of Fig. 2. An X indicates that no new raster number is computed; the one previously determined for that position stands. The letter L in a rectangle indicates that the display device is directed to draw a line between the points (I_1, J_1) and (I_2, J_2) currently specified. The complete process described by Fig. 4 is performed for each contour value prescribed.

Since several configurations have contours along the edges of the grid squares, it is conceivable that some contour segments might be drawn twice. For example, case Aθ could lie immediately above case Bϵ. In this eventuality the finished contour will merely have a segment

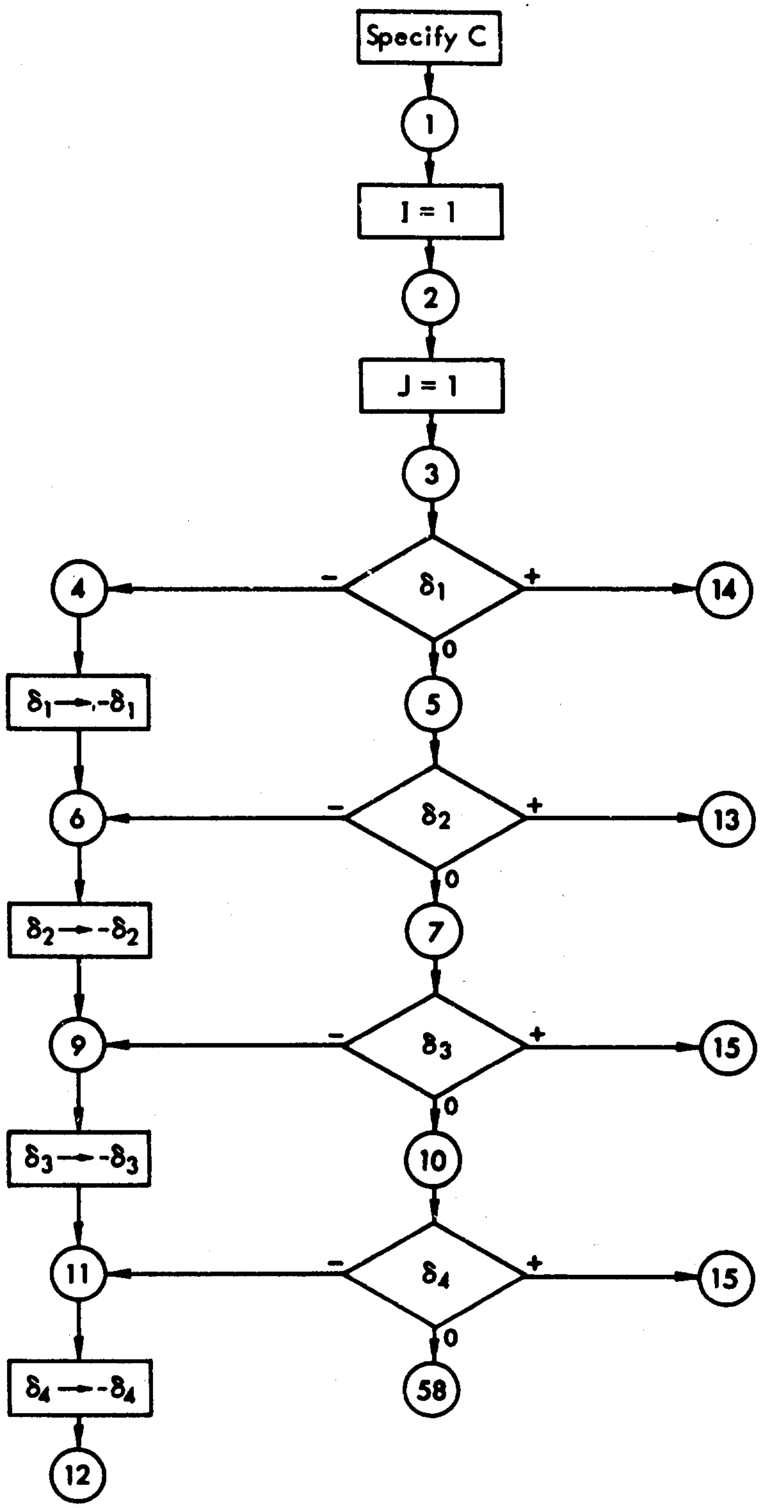

Fig. 4a -- Flow diagram.

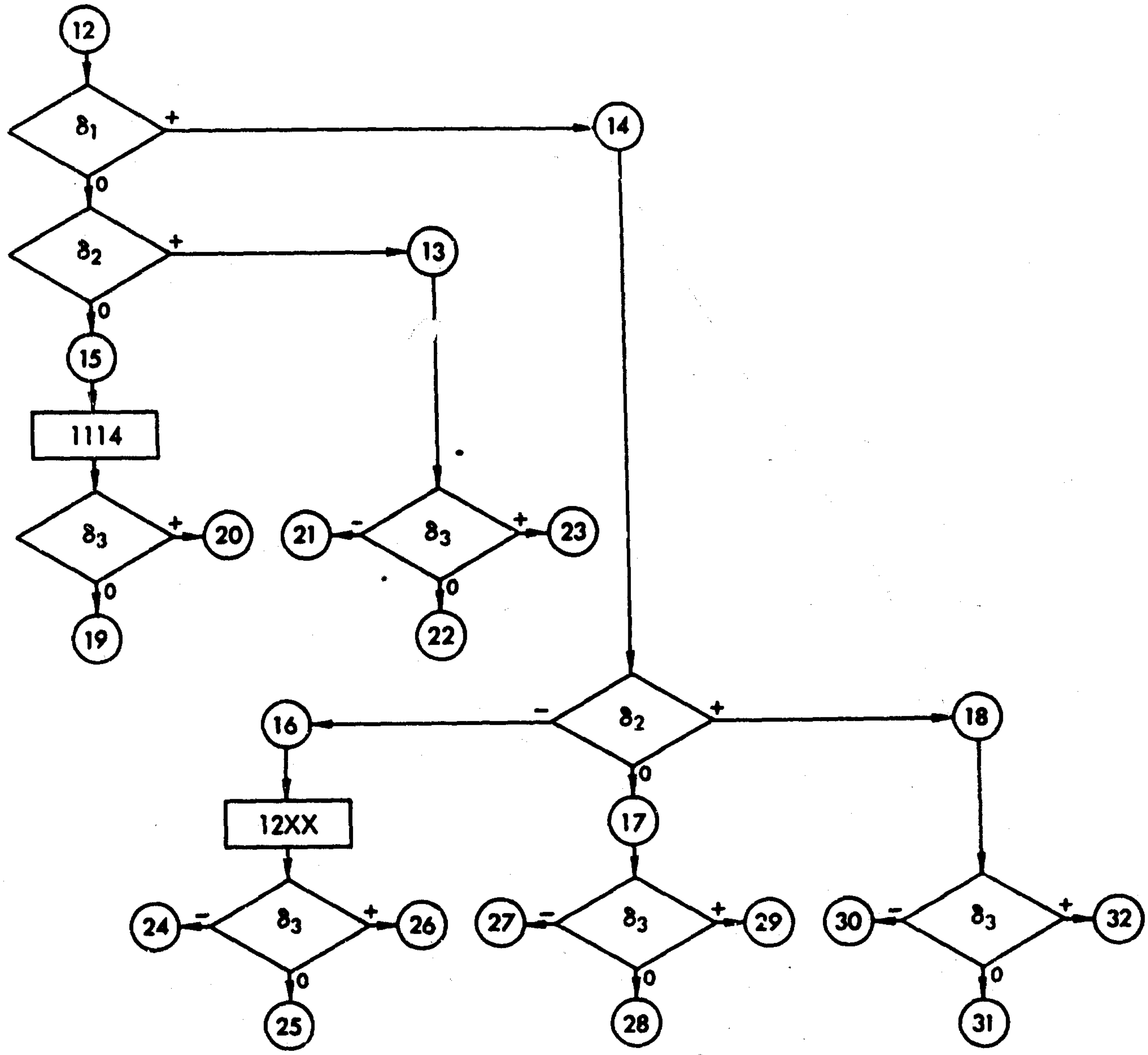

Fig. 4b -- Flow diagram.

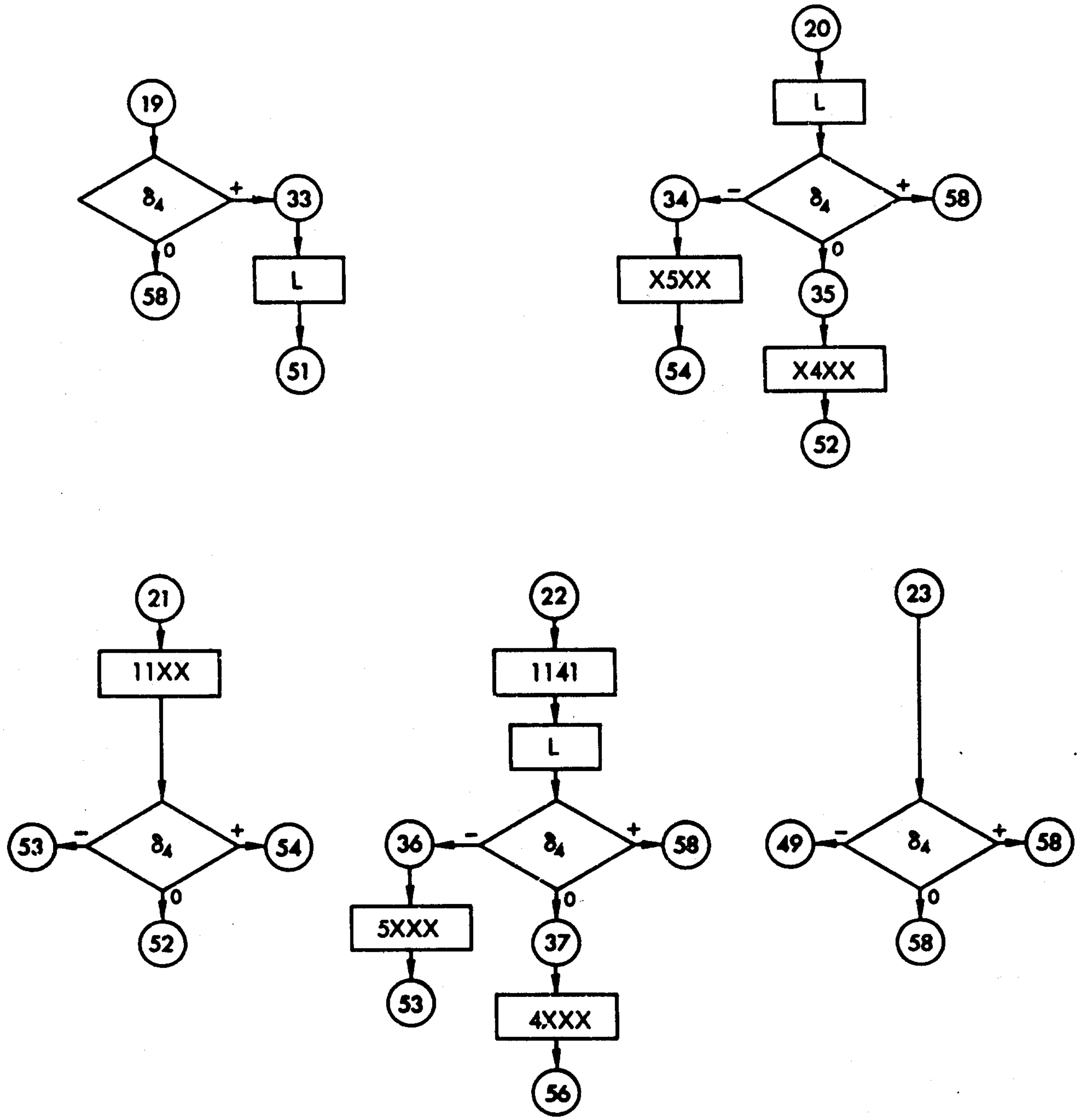

Fig. 4c -- Flow diagram.

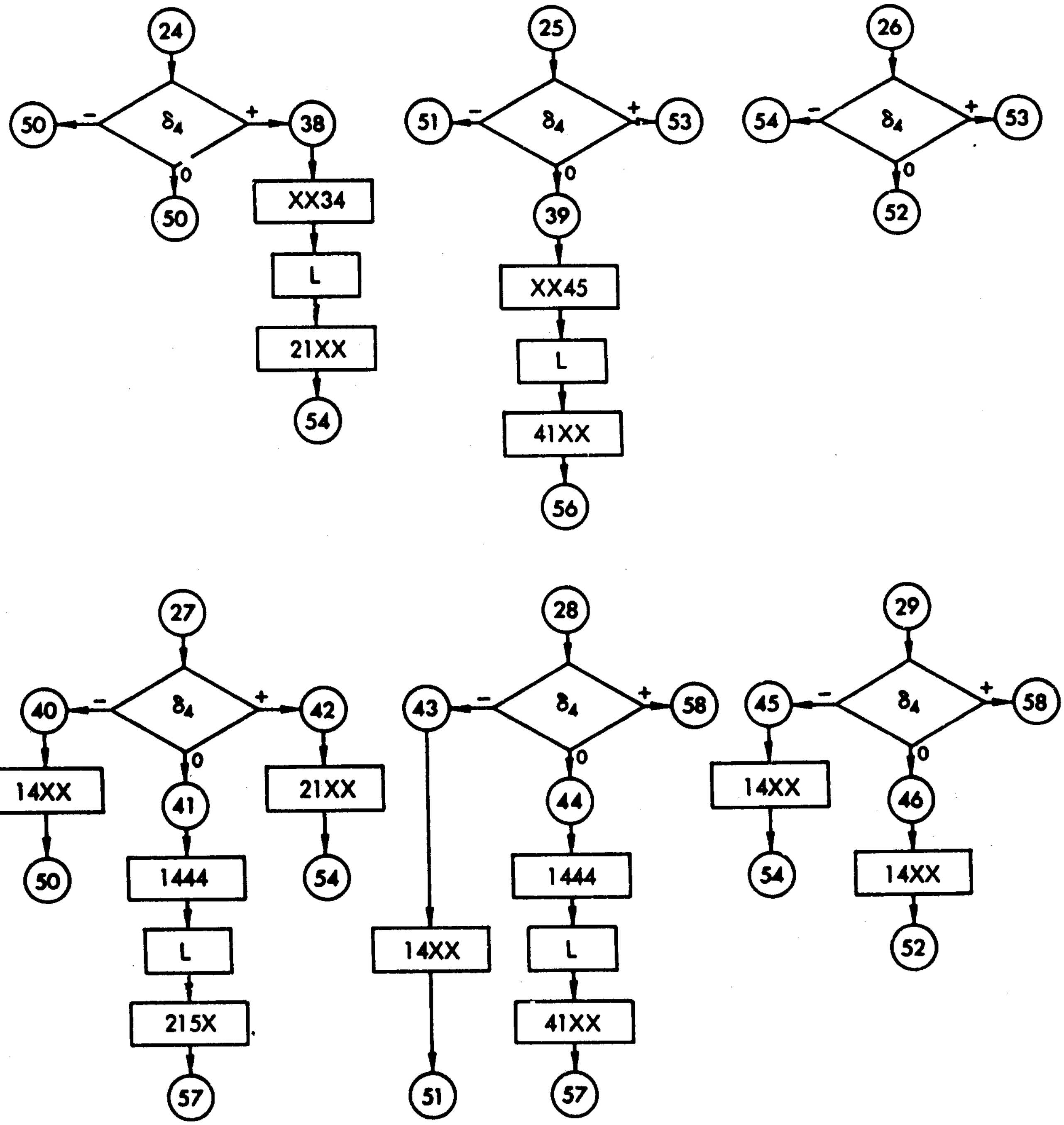

Fig. 4d -- Flow diagram.

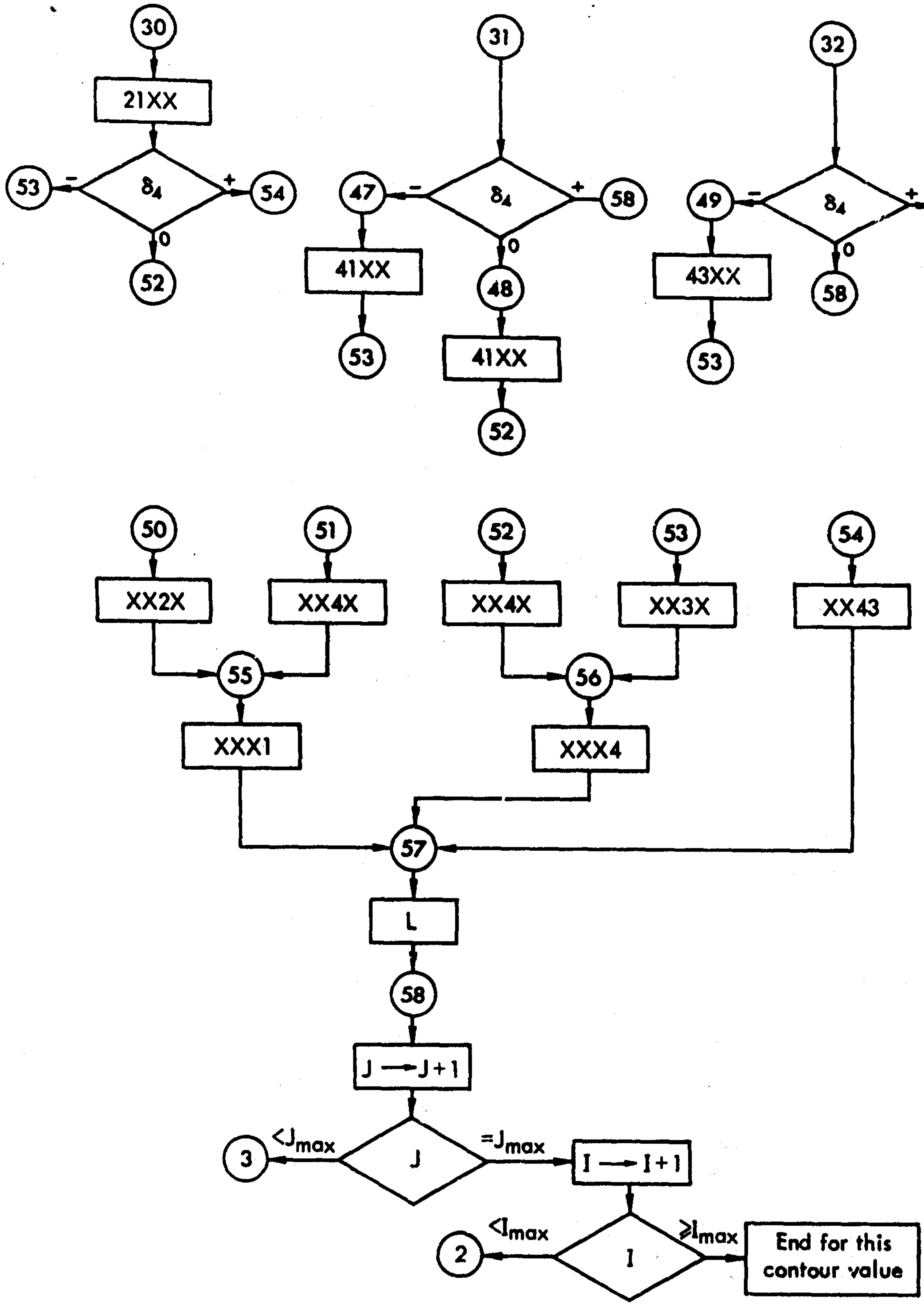

Fig. 4e -- Flow diagram.

darker than the rest of it.

Among the many possible juxtapositions of the 41 configurations of Fig. 2, some will occasionally produce contours different from those that would be hand-drawn by an analyst. One possibility is shown in Fig. 5. The analyst would probably draw some kind of nose or loop rather than the "dangling" contour between the squares labeled Dϵ and Bϵ, its exact shape depending on the magnitudes of δ_1 in the Dϵ square and δ_2 in the Bϵ square. For example, if the latter is near zero, the analyst might draw a curved contour segment (shown as a dashed line) in place of the segment common to the Bϵ and Aθ squares. In this case the objective program is concerned only with whether δ_2 of the Bϵ square is zero or nonzero, and ignores its magnitude. Numerous other cases could be constructed in which the objective contours have undesirable features, but in practice their occurrence will probably be infrequent.

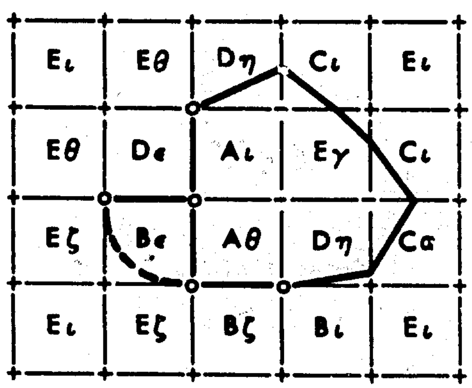

Fig. 5 -- Example of a "dangling" contour.

VI. ORIENTATION AND LABELS

Unless they are oriented with respect to a grid and identified in some way, the contours are of little value. The grid ordinarily consists of full lines entered at some interval prescribed by the programmer and labeled in the margin. If that set of lines, together with the contours, causes too much clutter, only the intersections of the grid lines need be designated, or, simplest of all, mere tick marks along the boundaries will sometimes suffice.

Most of the devices have means whereby numbers, letters, or other symbols can be displayed. This is important, as labels are essential. Among the labels required for almost any application are the following:

a. Title or other identification of the whole chart, including pertinent information such as date and time.
b. Grid labels sufficient to identify the coordinates of any point on the chart.
c. Indication of the value of G at significant points, such as maxima and minima.
d. Indication of the value of the contours shown.

Few difficulties are encountered in handling the first two items. The other two, however, are less simple. For example, how are the extrema of G to be defined? In the first place, we can distinguish between an absolute and a relative extremum. An absolute maximum, for example, could be a value that is not exceeded at any point in the domain, whereas a relative maximum could be a value that is not exceeded at any point in some neighborhood of the point in question. By these definitions the extrema are not unique; i.e., the extreme value can be shared by several adjacent points. A good example of this is a field that is positive in some limited region and zero elsewhere. In the large flat region of zeroes, each point is an absolute minimum, yet it would be most undesirable to label each as such.

The way chosen to solve this problem was to define as a relative maximum a grid-point value of G that exceeds the value at each of the eight nearest grid points (with the converse definition for a relative minimum), and to label only those absolute extrema that are also relative

extrema. To be sure, some bonafide relative extrema will be overlooked by this process (as where two adjacent grid points have the same extreme value), but this is thought to be preferable to the labeling of several adjacent points in a flat region.

Even so, if G does not vary smoothly, there may be many inconsequential relative extrema selected in a nearly flat region. It may not be desirable to reduce the incidence of these extrema by smoothing the field of G before contouring, yet the inclusion of the location and value of each of them would clutter the chart. One possible compromise is to indicate the location of each relative extremum with an appropriate symbol (as ⊞ for a maximum and ⊙ for a minimum), but to give value labels only to those relative extrema that are also absolute extrema or that differ in value from the nearest contour by more than a specified amount.

Finally, the value of the contours should be indicated. Ideally, each contour should be labeled at one point and at a few additional points if it is long or complicated in shape. A program that follows each contour from beginning to end can do this easily enough, but one that treats each contour as a number of short segments rather than as an entity cannot. It would be intolerable to label each segment, and complex and cumbersome to program logic to determine which one of the many segments constituting a contour should be labeled. Fortunately, the labeling of contours, although highly desirable, is not mandatory. If the values of a base contour and of the uniform contour interval are specified on the chart, together with the locations and values of important extrema, the value of any particular contour can be inferred by counting.

VII. EXAMPLES

The procedure described herein has been programmed for use with the General Dynamics S-C4060. Some examples of fields computed by a cumulus dynamics program are shown in Figs. 6, 7, 8, and 9.

At the top of each chart is a line describing the field and giving the simulated time or "cloud time" in minutes from the start of computation. On the same line the contour interval is shown. Although a base contour value and the contour interval are specified by the programmer, the program may multiply the specified interval by a power of 2 in order to avoid having too many or too few contours.

The grid is so designed that regardless of the mesh length of the given data, lines are printed and labeled at intervals of 1000 units. Additional boundary lines are also printed as needed. For example, in the data as presented to the contour program in the present example, grid points were at abscissae of -100, 100, 300, ..., 5500, 5700. Printed lines are at -100, 0, 1000, ..., 5000, 5700.

Maxima are indicated by the symbol $\boxed{+}$; minima by ⊙. They are labeled only if they are absolute extrema or if they differ from the nearest contour by more than a tenth of the contour interval. Contours are not labeled.

Figure 6 shows a slightly "rough" field; that is, one with numerous insignificant extrema and some contour irregularities. In the upper right-hand part of the chart are a "dangling" contour, some heavy, doubly struck segments, and other oddities. They occur in a relatively flat region, where the value of the dependent variable does not differ greatly from zero, which is also the value of the irregular contour. Slight smoothing before contouring can eliminate most of these irregularities, as is shown in Fig. 7. The original field portrayed in Fig. 6 was smoothed by substituting $(8\ G_{m,n} + G_{m+1,n} + G_{m-1,n} + G_{m,n+1} + G_{m,n-1})/12$ for $G_{m,n}$ at each interior point (m,n). The appearance of the contours improves somewhat in the upper right-hand part, but smoothing adversely affects the important small-scale patterns in the lower left. For example, the minimum of -3.87×10^{-2} on the left-hand boundary of Fig. 6 becomes a minimum of -2.81×10^{-2} in Fig. 7, and

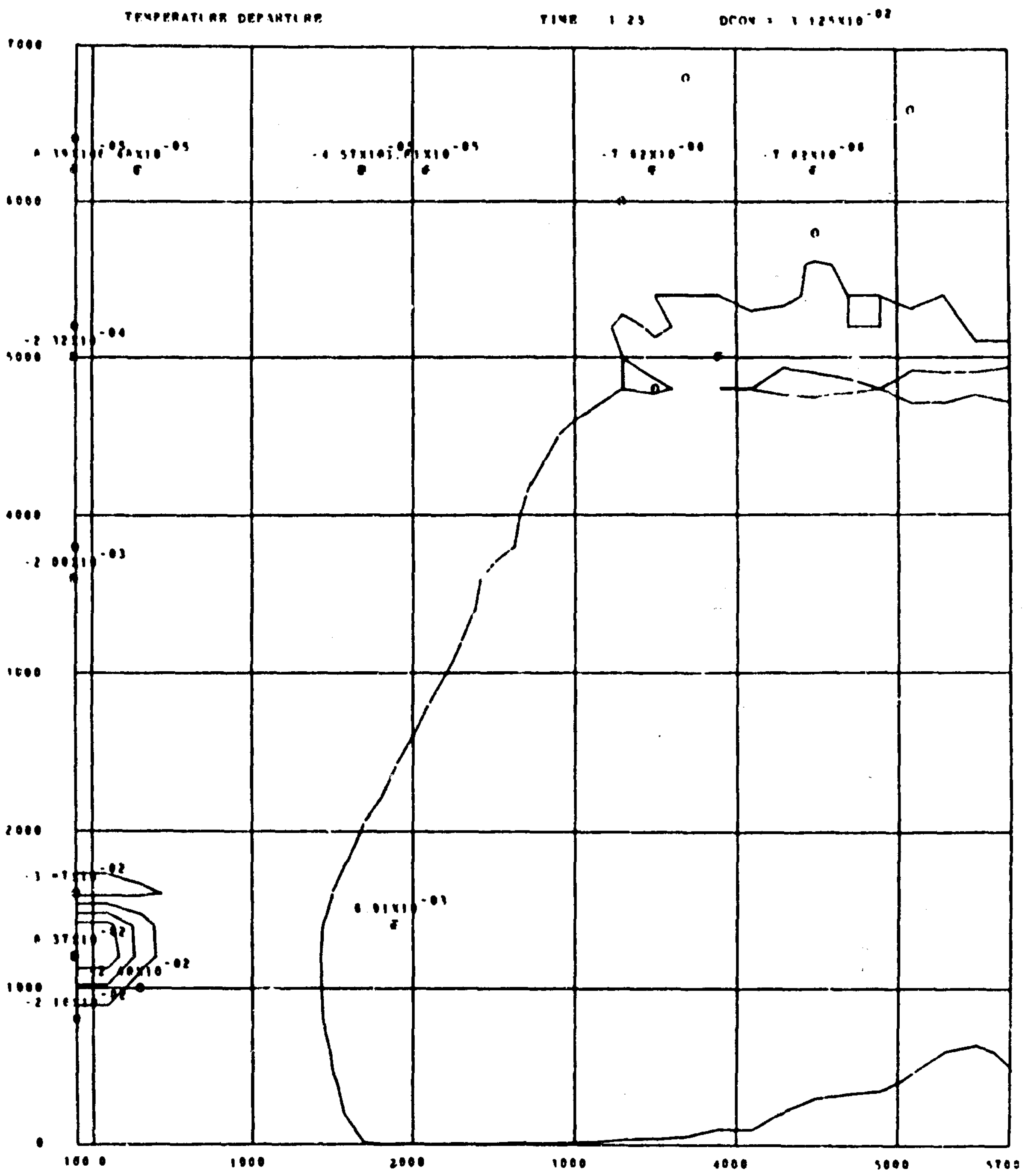

Fig. 6 -- Example of a slightly "rough" field.

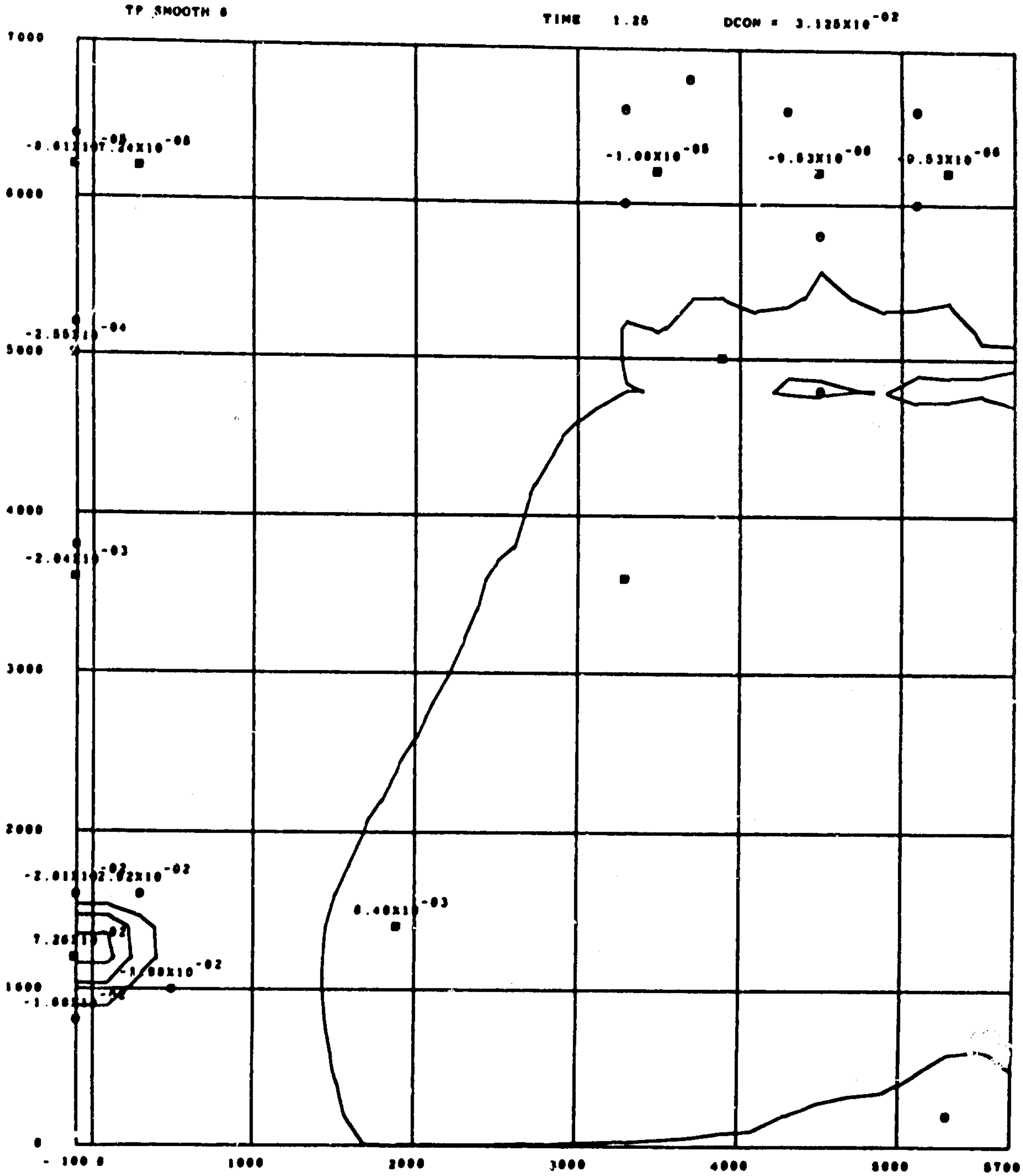

Fig. 7 -- Example of a smoothed field.

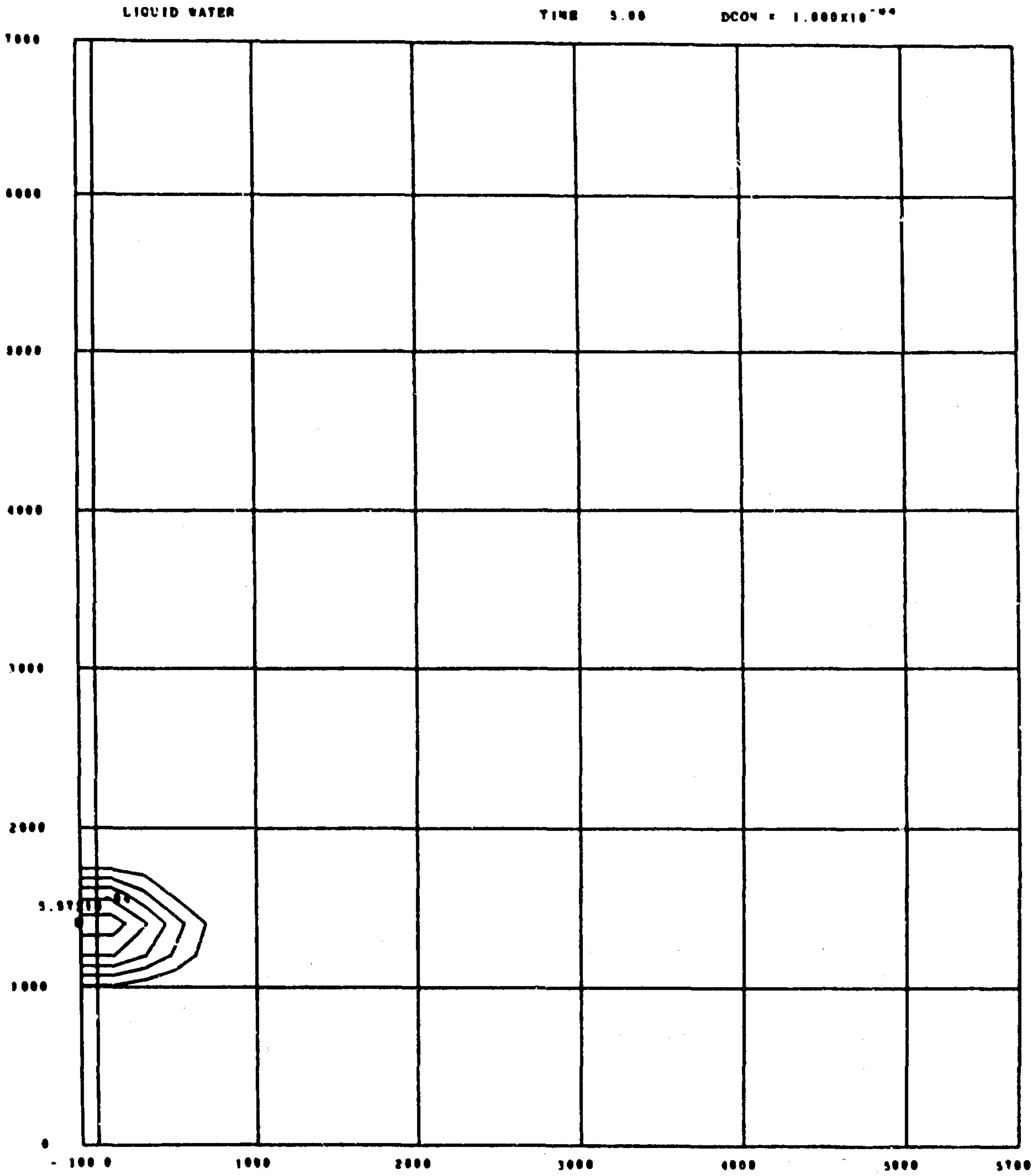

Fig. 8 -- Example of a smooth field with large flat region.

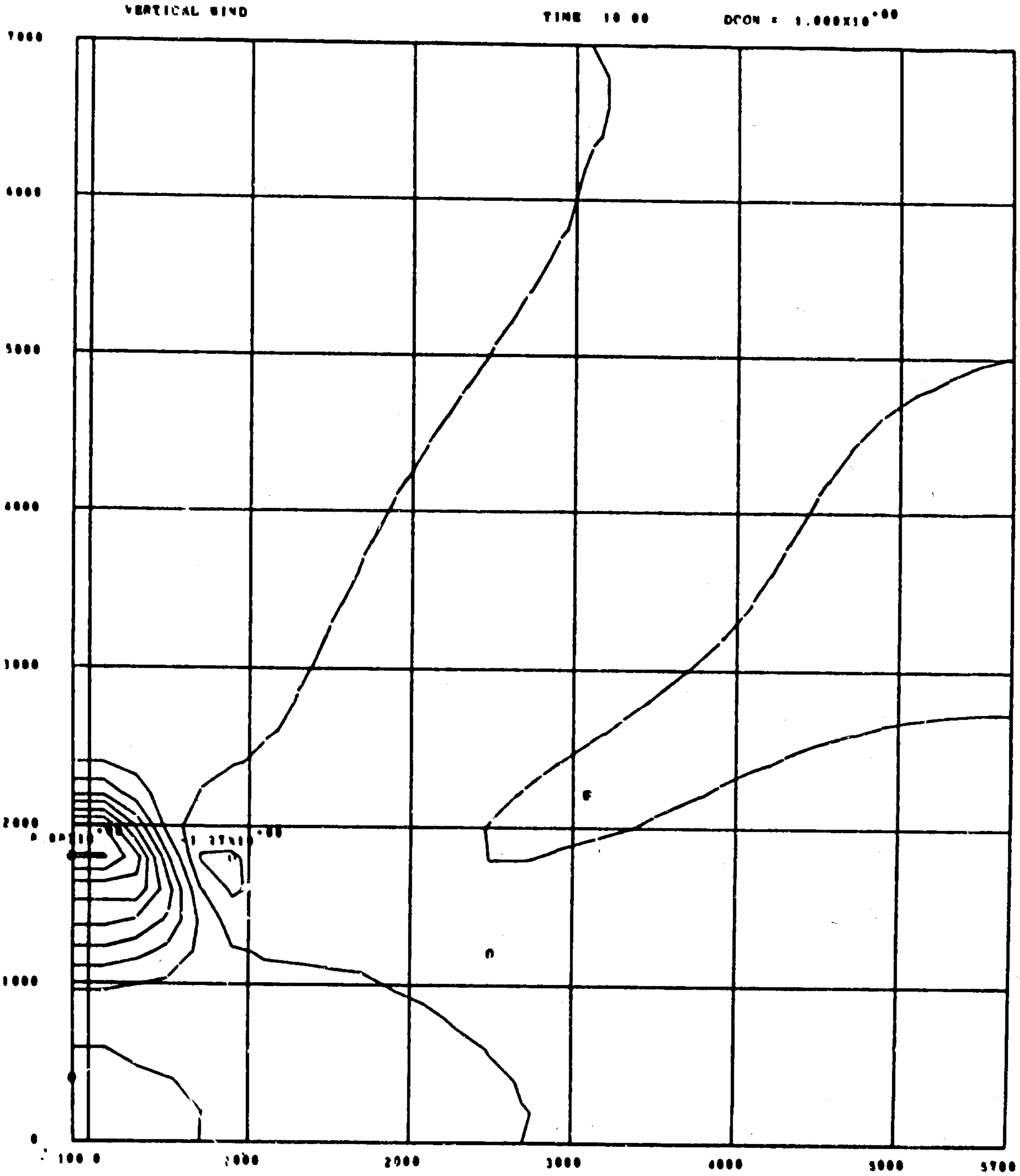

Fig. 9 -- Example of a complicated smooth field.

the contour of value -3.125×10^{-2} surrounding it vanishes. Hence this type of smoothing before contouring is not recommended for any field with small-scale features of interest.

Figure 8 depicts a field with no apparent irregularities. Over most of the area the value of the dependent variable is exactly zero, which is also the value of the outermost contour shown.

The field depicted by Fig. 9 is a little more complicated, but still smooth. There are three zero contours separating the two positive and two negative regions, identified by the symbols for maxima and minima.

The use of straight-line segments in these examples occasionally produces undesirable results, but on the whole the charts are acceptable. Certainly charts produced this way can be of great value in enabling one to study many fields without the slow and tedious work of transcribing and analyzing them by hand.

DOCUMENT CONTROL DATA

1 ORIGINATING ACTIVITY THE RAND CORPORATION	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

A METHOD OF OBJECTIVE CONTOUR CONSTRUCTION

4. AUTHOR(S) (Last name, first name, initial)

Murray, F. W.

5. REPORT DATE	6a. TOTAL No. OF PAGES	6b. No. OF REFS.
February 1968	32	---

7. CONTRACT OR GRANT No.	8. ORIGINATOR'S REPORT No.
N00014-67-C-0101	RM-5564-NRL

9a AVAILABILITY/LIMITATION NOTICES	9b. SPONSORING AGENCY
DDC-1	Office of Naval Research (Naval Research Laboratory)

10. ABSTRACT

A method for automatically constructing graphical contours of large numbers of data fields, such as those produced by many numerical models used in meteorology and other branches of geophysics. The data are most conveniently analyzed by drawing contours of the dependent variables on a grid representing two independent variables. Quite acceptable contours can be generated by RAND's simple general-purpose method, which uses the General Dynamics S-C 4060 computer-driven electronic graphical plotter. The computer (1) scans all the grid squares once for each value corresponding to a contour to be drawn, (2) finds, by linear interpolation, all the points where the contour intersects the edge of a grid square, and (3) instructs the graphical-output device to connect these points by straight lines. The program computes raster numbers for the two coordinates of each end-point of each segment; there are a total of 41 possible conditions, each requiring its own method of computation. The output is not drawn directly on paper but appears on the cathode ray tube display screen, where it can be photographed. A flowchart of the program logic and samples of the output are included.

11. KEY WORDS

Computer graphics
Curve fitting
Meteorology
Geophysics
Computer simulation
Clouds